Alien codes

Aliens have discovered that humans encode their messages as bit strings of 0s and 1s. They are eager to communicate with humans and would like to find how to encode their alien language as bit strings of 0s and 1s. They know that with a bit string of length N it is possible to have 2^N distinct bit strings. But the limitation of their signal transmission equipment does not allow transmission of bit strings with three consecutive 0s or two consecutive 1s. For example, it is not possible to transmit the following bit strings of length 10: 0101000101, 0101110010 or 1000101101. So they need to compute how many distinct bit strings there are, of a certain length N, with these constraints. How might you as a human solve this problem?

Input Format

Each test case consists of a single line containing an integer N indicating the length of the bit string to consider

Constraints

$4 \leq N \leq 64$

Output Format

A single integer being the number of bit strings of length N that do not have three consecutive 0s or two consecutive 1s.

Sample Input 0

```
4
```

Sample Output 0

```
5
```

Explanation 0

Out of the 16 possible distinct bit strings of length 4, only 5 of them can be transmitted because they have no three consecutive 0s and no two consecutive 1s. They are: 0010, 0100, 0101, 1001 and 1010

Sample Input 1

```
16
```

Sample Output 1

```
151
```

Array Combinations

Write a recursive Java program that reads from console a 2*n array where 2 is the number of rows and n is the number of columns. Your program should output all combinations of each subarray in order. The combinations should contain one element from every subarray.

Input Format

The first line represents the first subarray and the second line the second subarray. The elements of each subarray is seperated by commas.

Constraints

NA

Output Format

The output displayed on the screen is the combinations of each subarray in order.

Example

Input:

5

2,4

Output:

5 2

5 4

Another Example:

Input:

10,20

1

Output:

10 1

20 1

The Coin Change Problem

HackerRank

You have m types of coins available in infinite quantities where the value of each coin is given in the array $C = [c_0, c_1, \ldots, c_{m-1}]$. Can you determine the number of ways of making change for n units using the given types of coins? For example, if $m = 4$, and $C = [8, 3, 1, 2]$, we can make change for $n = 3$ units in three ways: $\{1, 1, 1\}$, $\{1, 2\}$, and $\{3\}$.

Given n, m, and C, print the number of ways to make change for n units using any number of coins having the values given in C.

Input Format

The first line contains two space-separated integers describing the respective values of n and m.
The second line contains m space-separated integers describing the respective values of $c_0, c_1, \ldots c_{m-1}$ (the list of distinct coins available in infinite amounts).

Constraints

- $1 \leq c_i \leq 50$
- $1 \leq n \leq 250$
- $1 \leq m \leq 50$
- Each c_i is guaranteed to be distinct.

Hints

- *Solve overlapping subproblems using Dynamic Programming (DP):*
  You can solve this problem recursively but will not pass all the test cases without optimizing to eliminate the overlapping subproblems. Think of a way to store and reference previously computed solutions to avoid solving the same subproblem multiple times.
- Consider the degenerate cases:
  - How many ways can you make change for 0 cents?
  - How many ways can you make change for > 0 cents if you have no coins?
- If you're having trouble defining your solutions store, then think about it in terms of the base case $(n = 0)$.
- The answer may be larger than a 32-bit integer.

Output Format

Print a long integer denoting the number of ways we can get a sum of n from the given infinite supply of m types of coins.

Sample Input 0

```
4 3
1 2 3
```

Sample Output 0

```
4
```

Explanation 0

There are four ways to make change for $n = 4$ using coins with values given by $C = [1, 2, 3]$:

1. $\{1, 1, 1, 1\}$
2. $\{1, 1, 2\}$
3. $\{2, 2\}$
4. $\{1, 3\}$

Thus, we print 4 as our answer.

Sample Input 1

```
10 4
2 5 3 6
```

Sample Output 1

```
5
```

Explanation 1

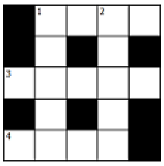
There are five ways to make change for $n = 10$ units using coins with values given by $C = [2, 5, 3, 6]$:

1. $\{2, 2, 2, 2, 2\}$
2. $\{2, 2, 3, 3\}$
3. $\{2, 2, 6\}$
4. $\{2, 3, 5\}$
5. $\{5, 5\}$

Thus, we print 5 as our answer.

Crossword counter

A crossword is a word puzzle that takes the form of a square or a rectangular grid of white and black shaded squares.



Given the layout of a crossword grid, you are asked to write a program to count the number of “across” words and the number of “down” words, knowing that a word is a sequence of at least 2 adjacent white squares that lie on the same horizontal or vertical line.

Input Format

Each test case consists of a single line containing two positive integers H and W indicating the height and width of the grid, followed by H lines, each line consisting of W characters describing the grid, where ‘.’ denotes a white square and ‘x’ denotes a black square.

Constraints

$1 \leq H \leq 100$
$1 \leq W \leq 100$

Output Format

For each test case, output one line containing the number of across words and the number of down words (separated by a single space) in the grid.

Sample Input 0

```
5 5
x....
x.x.x
.....
x.x.x
....x
```

Sample Output 0

```
3 2
```

Explanation 0

Grid of 5 lines and 5 columns
3 accross words
2 down words

Even Sheep

On a farm divided into a grid of cells, every cell either has grass on it or is empty.

If two adjacent cells have grass, they will belong to a common field. The common field extends in all directions to all adjacent cells with grass. So, if cell A is adjacent to cell B and cell B is adjacent to cell C, and all three have grass, then they all lie in the same field. If a cell with grass has no adjacent cell with grass, then it will be a field 1-cell field.

Every field must feed one sheep or one cow. Each field of grass cannot be shared between cows and sheep. If each field can have one sheep or one cow and never both, how many possible unique arrangements can you make such that, there are even number of sheep in the grid farm?

Input Format

The first line contains R (number of rows) and C (number of columns), separated by a space. Each of the next R lines contains a string with length equal to C, with no spaces. The string has the character Y to denote a cell with grass and N to denote a cell with no grass.

Constraints

1 ≤ R, C ≤ 5000

Output Format

S, an integer that contains the number of arrangements possible, modulo 1,000,000,007.

Sample Input

3 4
YNNY
NYNY
NYNN

Sample Output

4

Explanation

There are three fields, as follows:
1	-	-	2
-	3	-	2
-	3	-	-

First Solution (zero sheep)
Cow
Cow
Cow

Second Solution (two sheep)
Sheep
Cow
Sheep

Third Solution (two sheep)
Sheep
Sheep
Cow

Fourth Solution (two sheep)
Cow
Sheep
Sheep

So, the total number of ways is 4.

Flash disk storage

When using a USB flash disk as a storage device, it is beneficial to use as much space as possible, when saving files. You are asked to write a program to determine the most efficient way to store files onto a 2GB-flash disk (2048 MB), by using as much space as possible.

Input Format

Each test case consists of a single line containing an integer N indicating the number of files available to go onto the flash disk, followed by N lines, each line consisting of an integer S being the size in megabytes (MB) of each of the files.

Constraints

$1 \leq N \leq 20$
$1 \leq S \leq 2048$

Output Format

For each test case, output one line containing the sum (in MB) of the file sizes to be stored on the flash disk resulting in the most amount of space used.

Sample Input 0

```
5
1800
900
180
225
600
```

Sample Output 0

```
2025
```

Golf tournament

HackerRank

The MENA Golf Tournament was held in Saudi Arabia on October 5, 2016. Tournament participants were required to play 9 holes of golf and then submit their scores for each hole to the judges. The judges are still trying to determine the winners. Your job is to assist them in determining the top five winners in the golf tournament. The top five winners will be the participants who scored the lowest scores for the round of 9 holes of golf.

Input Format

The first line of input file will contain a positive integer N representing the number of participants in the tournament, followed by, for each participant, his full name on one line and, on the next line, his score S for each of the 9 holes, separated by a single space.

Constraints

$5 \leq N \leq 100$
$1 \leq S \leq 17$

Output Format

The output file will consist of five lines, each line consisting of the name and total score, separated by a single space, of the top five winners in the tournament.
Note: If two or more golfers have identical scores, list them in alphabetical order.

Sample Input 0

```
6
Rayan Thomas
1 2 3 4 5 6 7 8 9
Stephen Dodd
3 4 5 3 4 6 2 4 5
Joshua White
3 4 5 6 7 8 7 2 3
Zane Scotland
2 3 3 5 6 7 8 2 1
Matias Calderon
4 6 3 3 4 5 2 2 3
Abdulrahman AlMansur
5 6 4 6 6 7 8 3 4
```

Sample Output 0

```
Matias Calderon 32
Stephen Dodd 36
Zane Scotland 37
Joshua White 45
Rayan Thomas 45
```

Markov matrix

A square matrix of dimension N is called a positive Markov matrix if each element is positive and the sum of the elements in each column is equal to 1. You are asked to write a program to check whether a matrix is a positive Markov matrix or not.

Input Format

Each test case consists of a single line containing an integer N indicating the dimension of the square matrix, followed by N lines, each line consisting of N decimal numbers being the values of each row of the matrix.

Constraints

$2 \leq N \leq 25$

Output Format

For each test case, output **Yes** if the matrix is a Markov matrix, otherwise output **No**

Sample Input 0

```
3
0.150 0.875 0.375
0.550 0.005 0.225
0.300 0.120 0.400
```

Sample Output 0

```
Yes
```

Path crossover

Crossover is an operator used in genetic algorithms to change the programming of a chromosome from one generation to another. In this problem, you are asked to apply the crossover operator to find a new path out of two existing paths, in a path-planning algorithm.

A path, which represents a chromosome in genetic algorithms, is encoded as a list of cities.

For example, the following structure represents a possible path between cities: Riyadh->Jeddah->Madinah->Makkah->Taif->Abha

Let us assume that every city is represented by a number (or index). For example, if we assign these numbers to cities:
Riyadh = 1
Jeddah = 2
Makkah = 3
Madinah = 4
Abha = 5
Taif = 6

In this case, the path shown above will become: 1->2->4->3->6->5->1

Now, path crossover consists in taking two distinct paths and generating a new path using crossover.

The new third tour is created as follows:
- We select a subpath with length N starting from position K in Path1.
- Create Path3 using subpath so that subpath is located at the same indices in Path1
- Complete Path3 with remaining cities from Path2, in the same order as they appear in Path2.

Input Format

The input consists of 5 lines.
- the first line contains the length of the path, including the start and destination, which are the same city.
- the second line contains the position of the crossover
- the third line contains the length of the subpath
- the fourth line contains the first path
- the fifth line contains the second path

Constraints

Constraints on the path
To be valid, the path must satisfy the following constraints:
- The path starts and ends with the same city. In our example, the path starts and ends with 1.
- There is no city that repeats in the path, only the start is equal to the destination (first city and last city).
- The path must contain all the cities available (all numbers).
- The path must have at least 6 different cities.
- The position (index) of the first city is zero (0).

If any constraint above is violated, the program should return the string value **error** (case sensitive).

Output Format

The output will contain the path resulting from crossover

Sample Input 0

```
9
```

```
2
3
1 2 4 8 3 6 5 7 1
1 6 7 3 8 5 4 2 1
```

Sample Output 0

```
1 6 4 8 3 7 5 2 1
```

Explanation 0

To illustrate the process, consider the following example. We have two paths:
Path1: 1->2->4->8->3->6->5->7->1
Path2: 1->6->7->3->8->5->4->2->1

Now, we will make a crossover at position K=2 with length N=3 in Path1. So, the subpath is 4->8->3 because city 4 is at position 2 in Path 1 and the length of the subpath is 3 (cities 4 and 8 and 3).

Therefore, the third path will be initialized as follows:
Path3: 1->?->4->8->3->?->?->?->1

This path must be completed from Path2 in the same order of cities.

In Path2, the first element after the start city is 6. As city 6 does not belong to the subpath, we will add it to Path3.
Path3: 1->6->4->8->3->?->?->?->1

The next value in Path2 is city 7 and is not in subpath, so we add city 7 to Path3.
Path3: 1->6->4->8->3->7->?->?->1

The next value in Path2 is city 3. However, city 3 already exists in subpath, so we skip it.

The next value in Path2 is city 8. However, city 8 already exists in subpath, so we skip it.

The next value in Path2 is city 5 and is not in subpath, so we add city 5 to Path3.
Path3: 1->6->4->8->3->7->5->?->1

The next value in Path2 is city 4. However, city 4 already exists in subpath, so we skip it.

The next value in Path2 is city 2 and it is not in subpath, so we add city 2 to Path3.
Path3: 1->6->4->8->3->7->5->2->1

This is Path3, the resulting path from crossover between Path1 and Path2.

Sample Input 1

```
10
2
5
3 4 8 9 2 6 5 7 1 3
3 8 7 5 9 1 2 5 4 3
```

Sample Output 1

```
error
```

Explanation 1

In Path2, 5 is repeated

Pattern Recognition

In image processing, is the process of classifying input data into objects or classes based on key features.

In this problem, you will develop a pattern recognition algorithm that identify how many classes exists for a certain pattern in an image.

First, an image is considered as a 2D array of integer values, where each element in the matrix has a value between 0 and 255. If any of the element is not in this bound, the program should terminate with the integer -1. The objective is to find how many times a pattern repeats with a certain similarity index. The similarity index is a value between 0.1 and 1 (inclusive), which represent the similarity between a pattern and certain elements in the matrix. If any of the similarity index is not in the [0.1, 1], the program should terminate with the integer -1.

Let is consider this example. Consider the following image with 7x7 size.

1	54	8	0	0	255	255
22	3	41	102	145	178	141
3	24	78	58	96	74	52
5	122	145	32	3	41	78
4	11	21	54	24	78	12
3	41	54	54	98	45	14
24	78	251	240	93	201	155

Note the size of the matrix can be any number N>0.

Now, consider the following pattern with 2x2 size that we look for

3	41
24	78

Note the size of the pattern can be any number P

So, in this case, if we consider a similarity index of 1, then, we have three classes of this pattern in this image, which are:

1	54	8	0	0	3	255
22	**3**	**41**	102	145	24	78
3	**24**	**78**	58	96	74	52
5	122	145	32	**3**	**41**	78
4	11	21	54	**24**	**78**	12
3	**41**	54	54	98	45	14
24	**78**	251	240	93	201	155

A similarity index of 1 means that there must be 100% similarity between the patterns and the elements of the classes.

Now, for the same image and the same pattern if the similarity index is 75%, in this case, we have then four classes of this pattern in this image, which are

1	54	8	0	0	3	255
22	**3**	**41**	102	145	24	78
3	**24**	**78**	58	96	74	52
5	122	145	32	**3**	**41**	78
4	11	21	54	**24**	**78**	12
3	**41**	54	54	98	45	14
24	**78**	251	240	93	201	155

The fourth class is considered as a pattern matching because 3 cells out of 4 are the same as in the pattern, so a similarity of ¾=0.75. So, for any similarity index greater than 0.75 up to 1 we will have in this particular case three classes, and for a similarity index lower than 0.75 we will have at least four classes.

If the similarity index was 0.5, we will also count as pattern matching the case where two elements (or more) out of four in the pattern matches elements in the image. A pattern must be counted only once in case of a similarity lower than 1.0.

Sample input:
7 7
1 54 8 0 0 3 255
22 3 41 102 145 24 78
3 24 78 58 96 74 52
5 122 145 32 3 41 78
4 11 21 54 24 78 12
3 41 54 54 98 45 14
24 78 251 240 93 201 155
1.0
2 2
3 41
24 78

Sample output:
3

Explanation: The input will contain the following lines. The first line will contain the size of the matrix, where the first int is the number of lines, and the second in is the number of columns. The next lines will contains the image matrix values. After the image elements are read, the next line contains the similarity index. Then, the subsequent lime contains the size of the pattern. The next lines will then contains the values of pattern elements.

The output will contain number of classes found for the specified similarity index.

Input Format

The input contains the following The first line will contain the size of the matrix, where the first integer is the number of lines, and the second integer is the number of columns. The next lines will contains the image matrix values. After the image elements are read, the following line contains the similarity index. Then, the subsequent lime contains the size of the pattern (rows and columns). The next lines will then contains the values of pattern elements.

```
7 7
1 54 8 0 0 3 255
22 3 41 102 145 24 78
3 24 78 58 96 74 52
5 122 145 32 3 41 78
4 11 21 54 24 78 12
3 41 54 54 98 45 14
24 78 251 240 93 201 155
0.74
2 2
3 41
24 78
```

```
In the above a 7x7 matrix is provided
1 54 8 0 0 3 255
22 3 41 102 145 24 78
3 24 78 58 96 74 52
5 122 145 32 3 41 78
4 11 21 54 24 78 12
3 41 54 54 98 45 14
24 78 251 240 93 201 155

The similarity index is 0.74

The pattern to match is a 2x2 matrix
3 41
24 78
```

Constraints

Number of rows and columns for the matrix and the pattern must be >0. Similarity index cannot be smaller than 0.1 All matrix values must be >=0 and <=255

Output Format

The output will contain number of classes found for the specified similarity index. For the above example, output is 4

For any unacceptable input or wrong solution and program outputs -1.

Sample Input 0

```
7 7
1 54 8 0 0 3 255
22 3 41 102 145 24 78
3 24 78 58 96 74 52
5 122 145 32 3 41 78
4 11 21 54 24 78 12
3 41 54 54 98 45 14
24 78 251 240 93 201 155
1.0
2 2
3 41
24 78
```

Sample Output 0

```
3
```

Sample Input 1

```
7 7
1 54 8 0 0 3 255
22 3 41 102 145 24 78
3 24 78 58 96 74 52
5 122 145 32 3 41 78
4 11 21 54 24 78 12
3 41 54 54 98 45 14
24 78 251 240 93 201 155
0.74
2 2
3 41
24 78
```

Sample Output 1

```
4
```

Permuted words

A word can be permuted by changing its letters positions. For example, **[own, won, now]** are all generated by changing the letters positions. This has many applications especially in security. You are asked to develop a program that collects permuted versions of a word. For any given word, your program will check if there exist any of its permutations and if so it will display the number of available permutations for it. If the word itself is a new permutation and it does not already exist, the program would add the word to the stored set of permutations. Your program should be case insensitive (**uppercase and lowercase letters are considered the same**).

Input Format

Each test case consists of a single line containing one positive integer N indicating the number of words to process, followed by N words, each on a separate line

Constraints

$1 \leq N \leq 25$

Output Format

For each word in the test case, output one line containing the word as is, the number of already added permutations and whether the word was added or not (Y or N), all separated by one space

Sample Input 0

```
6
now
won
CAT
WON
owN
ACT
```

Sample Output 0

```
now 0 Y
won 1 Y
CAT 0 Y
WON 2 N
owN 2 Y
ACT 1 Y
```

Semantic Mind-Reader

In the race for the best Internet browser, there is now a new contender called the The Semantic Mind-Reader. After its promo on the world wide web, everyone has been desperately waiting for the browser to be released. Apart from the various security powers it possesses, it is called the mind-reader for a reason. You do not need to type 'www.' to open a website anymore. Though, you still need to type '.com' to open a website. The browser predicts all the vowels in the name of the website. Obviously, this means you can type the name of a website faster and save some time. To convince the users that this browser will indeed save them a lot of time to open a website, you are asked to write a marketing report.

Input Format

Each test case consists of a single line containing an integer N followed by N lines, each line consisting of a a string of length L being a website address in lowercase letters.
Note: Every website address starts with **www.** and ends with **.com**

Constraints

$1 \leq N \leq 100$
$1 \leq L \leq 200$

Output Format

The ratio of the number of characters you would type in the new browser, to the number of characters you would have typed in your normal browser.

Sample Input 0

```
2
www.google.com
www.hackerrank.com
```

Sample Output 0

```
7/14
11/18
```

Explanation 0

In the first test case, you would type **ggl.com** (7 characters only) whereas in a normal browser, you would type **www.google.com** (14 characters)

Social Relationship Analyzer

Bob has been using facebook for many years; he has a habit of friending anyone who requests to be a friend. This is a dilemma, over the years, he has been in a relationship for over 5000 friends, many of them he has never communicated over the months.

He is thinking of building a fast program to filter unwanted friends based on their trustfulness. You need to help him build a fast program that would require minimum number of computations. Bob uses a strange method of defining his social relationships using a 2x2 matrix composed of trust relationships. He defines 3 states of relationships, trustful (represented by 1), so so (represented by 0) and not so trustful (represented by -1) in the matrix. These matrices may not be necessarily be square matrices (i.e. rows and columns can be of different sizes).

	Bob	Alice
Bob	1	1
Alice	0	1
John	-1	1
Ahmed	-1	0

Since he has over 5000 friends, he cannot afford to have a very large matrix containing trust parameters. To determine an optimal solution, he has decided to build many small 2x2 matrices by asking his friends to provide trust information about their friends in the same manner. Luckily his friends always provide their matrices in a particular order:

A1(a x b), A2(b x c), A3(c x d) …. Where (a x b) represent dimensions of matrices (a rows and b columns) sent by Bob’s friends A1, A2 and A3… An.

Bobs algorithm computes trust scores but multiplying all of these matrices. Unfortunately multiplying all matrices take a long time and Bob is tired of waiting. He wants you to help him optimize his matrix multiplication problem by selecting an optimal order of trust matrix multiplication. He wants to make sure that any solution you provide will give the minimum number of multiplications (cost).

The cost of multiplying two matrices with dimensions 3x4 and 4x5 = 3 * 4 * 5 = 60. In multiplication of more than 2 matrices, the order of multiplication matters in determining the optimal minimum cost.

Input Format

The first line consists of a positive integer n > 0 representing number of matrices. In n subsequent lines, each line consists of dimensions of each matrix, number of rows followed by number of columns. All values for rows and columns must be >0.

5
4 10
10 3
3 12
12 20
20 7

Constraints

All values for rows and columns must be >0.

Output Format

Integer value >0 giving minimum number of computations in the optimal solution. The program writes -1 to console if any error occurs, i.e. if any number of rows or columns <=0 or the matrices cannot be multiplied.

Sample Input 0

```
5
4 10
10 3
3 12
12 20
20 7
```

Sample Output 0

```
1344
```

Explanation 0

The first row indicates number of matrices to be read from subsequent lines Each line contains a matrix with number of rows and columns

Verification

HackerRank

The current cloud technology is based on distributed networks which consist of supercomputers assembled in closely connected networks. Data can be shared and processed on a larger network and for cheaper, by letting individual computers contribute over a bigger network through cloud platform. Since individual clusters or nodes are someone else's computer that run untrusted code, the results are also not trusted, so some verification needs to take place to check that the work performed is legal and complies with the policies.

Input Format

Each test case consists of an integer indicating the number S of sets, followed by, for each set, an odd integer N indicating the number of workers followed by N integers indicating the results R for the same piece of work from each of the N workers

Constraints

$2 \leq S \leq 6$
$1 \leq N \leq 7$
$0 \leq R \leq 100$

Output Format

Will consist of S lines, each line indicating the verification about the performed work for each set. If the majority of the results agree (that is, they are the same), then the work is **Verified**. If there is no major consensus, but one result has more matches than any other result, then it is **Unverified**, otherwise the result is **Unknown**.
Note: a single result is always **Verified**.

Sample Input 0

```
3
3
80
22
80
5
10
15
10
25
30
3
50
52
51
```

Sample Output 0

```
Verified
Unverified
Unknown
```